

all components **108** installed with the software product **106** along with a list of changes that have been made to the software product **106**.

[0028] The system of **FIG. 1** is next described in greater detail.

[0029] Componentized Software Product

[0030] Referring next to **FIG. 2**, a block diagram illustrates an exemplary computing device **201** in accordance with certain embodiments of the invention. The computing device **201** is illustrated as having a software product such as operating system **202** and one or more application programs **204**.

[0031] Operating system **202** is the fundamental software control program for computing device **201**, performing various functions including providing a user interface, managing the execution of one or more applications **204**, and controlling the input of data from and output of data to various input/output (I/O) devices. Application programs **204** represent one or more of a wide variety of software application programs that may be executed on computing device **201**. Examples of such application programs **204** include educational programs, reference programs, productivity programs (e.g., word processors, spreadsheets, databases), recreational programs, utility programs (e.g., communications programs), etc. Application programs **204** may be installed on computing device **201** by the user, or alternatively pre-installed by the manufacturer and/or distributor of computing device **201**.

[0032] Operating system **202** separates its functionality into multiple components **206** such as component #1 through component #N. Each component **206** has a corresponding manifest **208** such as manifest #1 through manifest #N, respectively. The components **206** include a collection of one or more files (or file identifiers). The files may include software instructions such as an executable file, a dynamic-link library (DLL), or a component object module (COM). The files may also include data for use by one or more of the components **206**. In one implementation, the files (e.g., data and/or instructions) corresponding to particular functionality of the operating system **202** are grouped together in the same component **206**. For example, there may be a games component, a communications component, and a file system component. The grouping of files that result in the componentization may be static or alternatively may change over time. In one example, updates to operating system **202** may result in selected files from certain components **206** being removed and added to other components **206**.

[0033] Each manifest **208** includes information describing the corresponding component **206**. Any of a wide variety of metadata regarding the corresponding component **206** may be included in each manifest **208**. In one implementation, the manifest **208** identifies the version of the corresponding component **206** as well as which other components **206**, if any, the corresponding component **206** is dependent on. By way of example, in order for one or more files in component **206** to properly execute, one or more other files (e.g., a DLL file) from another component may need to be installed on computing device **201**. In this example, manifest **208** would indicate that component **206** depends on the other component.

[0034] In one form, one or more computer-readable media associated with computing device **201** have computer-ex-

ecutable modules for updating the software product **106** (e.g., operating system **202**) with a service package such as service package **104**. The computer-executable modules are embodied as a component installer **210**. The component installer **210** includes a configuration module **212**, an installation module **214**, a script module **216**, a report module **218**, and a dependency module **220**. The configuration module **212** determines the state associated with the component **206** to be updated and selects one of the instruction sets based on the determined state. The installation module **214** modifies the component **206** by applying one or more of the files to the component **206** in accordance with the selected instruction set. The script module **216** updates the manifest **208** for the component **206** with data related to applying the one or more of the files. The script module **216** further stores the updated manifest **208** for the component **206** with component **206**. The report module **218** receives a request from a user for data in the manifest **208**. The report module **218** further queries the manifest **208** in response to the received request to generate query results and provides the query results to the user. The dependency module **220** resolves dependencies between files in the service package and the component **206** in the software product. Those skilled in the art will note that the modules **212-220** may all be part of the component installer **210** or may be stored in separate application programs or any combination in-between. Further, the modules **212-220** may execute locally or remotely to provide the functionality. The component installer **210** may include additional or less functionality than illustrated and is described herein.

[0035] Updating a Component Software Product

[0036] Referring next to **FIG. 3**, a block diagram exemplifies how a componentized software product may be updated in accordance with certain embodiments of the invention. A computer such as computing device **201**, as discussed above with reference to **FIG. 2**, is illustrated including the componentized operating system **202**. A service package **302** is coupled to computing device **201** via a coupling such as network **310**. Service package **302** may include any of a variety of sources such as a magnetic disk, an optical disk, flash memory, and a solid-state disk. Additionally, service package **302** may be local (e.g., a disk that is accessed by a disk drive of computing device **201**) or remote (e.g., a disk at a remote server accessed over network **310**). Those skilled in the art will note that the exact nature of the coupling between the service package **302** and computing device **201** is dependent on the nature of service package **302** (e.g., the coupling may include a bus connecting a disk drive to computing device **201**, a modem and telephone line, a network adapter and network, or a combination of any of these).

[0037] Service package **302** includes a plurality of installation instruction sets **304**. As described previously, each of the instruction sets **304** corresponds to a state or phase of the components to be updated (e.g., components **206**) in the software product lifecycle. The instruction sets **304** include computer-executable instructions (e.g., software code or scripts) or declarative installation steps which are followed by an application program such as component installer **210** in **FIG. 2** during the update process. The service package **302** further includes updated components **306** such as updated component #1 through updated component #M. The service package **302** stores updated manifests **308** such as